



# **TECNICAS HEURISTICAS Y METAHEURISTICAS PARA EL PROBLEMA DE LA MAXIMA DIVERSIDAD (MAXIMUM DIVERSITY PROBLEM (MDP))**

Desarrollo y análisis de dos  
enfoques de tratamiento del  
problema MDP

Alumno: Miguel Ángel Franco Gálvez

Director: Domingo Giménez Cánovas

Septiembre 2015



# SUMARIO

- Introducción
- Estado del arte
- GRASP\_M
- EMP
- Conclusiones y trabajos futuros

# INTRODUCCIÓN

- Definición del problema MDP de Kuo, Glover y Dhir (1993):

$$\text{Maximizar} \quad \sum_{i=1}^{n-1} \sum_{j=i+1}^n d_{ij} x_i x_j$$

$$\text{Condicionado a} \quad \sum_{i=1}^n x_i = m$$

$$\text{con} \quad x_i = \{0, 1\} \quad 1 \leq x_i \leq n$$

$$\text{donde} \quad d_{ij} = \sqrt{(v_i - v_j)^2} = |v_i - v_j|$$

# INTRODUCCIÓN

- Complejidad del problema:
  - MDP es NP-HARD (Kuo, Glover y Dhir (1993))
  - Ejemplos:
    - $n = 10, m = 3: \binom{10}{3} = 120$
    - $n = 500, m = 50: \binom{500}{50} \approx 10^{69}$
  - Tamaño del espacio de soluciones inabordable mediante métodos exactos.
  - Utilización de heurísticas y metaheurísticas.



# INTRODUCCIÓN

- Aplicaciones prácticas MDP:
  - Localización de unidades logísticas.
  - Ubicación de instalaciones peligrosas.
  - Composición de jurados públicos en juicios.
  - Generación de nuevos medicamentos.
  - Políticas de inmigración.
  - ...



# INTRODUCCIÓN

- Motivación y objetivos:
  - Analizar importancia MDP.
  - Analizar estado del arte en resolución MDP.
  - Crear nuevo algoritmo desde cero (GRASP\_M).
  - Analizar bondad (fitness) del nuevo algoritmo.
  - Analizar metaheurísticas y algoritmo *EMP*.
  - *Conclusiones y trabajos futuros.*



# Estado del arte

## ■ Métodos exactos.

- Solución exacta.
- Poca eficiencia en general.

## ■ Métodos heurísticos.

- Solución factible generalmente no muy próxima al óptimo.
- Eficiencia media en general.

## ■ Métodos metaheurísticos.

- Solución factible generalmente próxima al óptimo.
- Eficiencia buena en general.

# Estado del arte

## ■ Métodos exactos.

### □ “Branch and Bound”

- Intento eliminar ramas exploración (prune) y acelerar resultados.
- Importancia “cotas” para buen funcionamiento.

### □ “Backtracking”. Variable Width Backtracking (VWB).

- Basado en refinamiento sucesivo de cotas en Backtracking.
- Posible alternativa para la generación de buenas soluciones iniciales.



# Estado del arte

- Heurísticas.

- Cierta inteligencia simple e intuitiva.
- No hay memoria sobre la búsqueda realizada.
- Objetivo: no tener que explorar el espacio de soluciones completo.

- Tipos.

- Constructivos.
- Búsqueda local o mejora de soluciones.
- Combinación de ambos.



# Estado del arte

- Metaheurísticas (I):
  - Mayor nivel de inteligencia que heurísticas.
  - Aplicables a varios tipos de problemas.
  - Pueden estar basados en observaciones del mundo real o campos teóricos (naturaleza, inteligencia artificial, estadística,...).



# Estado del arte

- Metaheurísticas (II):

- Ejemplos:

- GRASP.
    - SA (Simulated annealing).
    - TS (Tabu Search).
    - SS (Scatter Search).
    - PR (Path Relinking).
    - VNS (Variable Neighborhood search).
    - Otros (Genetic Algorithm, Memetic Algorithms, Ant colony...).

# GRASP\_M

- Principios de diseño:
  - Crear algoritmo de resolución MDP desde cero.
  - En principio, no tener en consideración nada de lo hecho, sobre todo procesos.
  - Desarrollo guiado por intuición y experimentación a partes iguales.
  - Énfasis en la optimización de datos y procesos.

# GRASP\_M

- Características principales:
  - Algoritmo tipo GRASP (por parecido).
  - Evolucionar una solución inicial semi-aleatoria guiada por la función objetivo.
  - Aplicación mejora intensa y semi-aleatoria.
  - Multiarranque.
  - Utilización cierta información de mejores soluciones obtenidas.

# GRASP\_M

## ■ El algoritmo:

```
1 SolucionActual = SolucionMejora = Vacia;
2 SolucionPropuesta = SolucionMejoraTemp = Vacia;
3 ConjuntoSoluciones = Vacio;
4 PrepararProblema(P);
5 TiempoAlgoritmo = 0;
6 MIENTRAS (TiempoMaximoAlgoritmo > TiempoAlgoritmo) {
7     SolucionPropuesta = GenerarNuevaSolucion(P, SolucionPropuesta);
8     SI ((SolucionPropuesta > SolucionActual) || CriterioMejora()) {
9         SI (SolucionPropuesta > SolucionActual)
10            SolucionActual = SolucionPropuesta;
11        SolucionMejoraTemp = MejorarSolucion(SolucionPropuesta);
12        SI(SolucionMejoraTemp > SolucionMejora) {
13            SolucionMejora = SolucionMejoraTemp;
14            iteracionesSinMejora = 0;
15        SI NO
16            IteracionesSinMejora = IteracionesSinMejora + 1;
17        SI (CriterioCambioSolucionPropuesta())
18            SolucionPropuesta = SolucionMejora;
19        SI NO SI (CriterioReinicio()) {
20            IncluirSolucionEnConjunto(ConjuntoSoluciones, SolucionMejora);
21            SI (CriterioReinicioConElementos())
22                SolucionPropuesta = SeleccionarElementos(ConjuntoSoluciones);
23            SI NO
24                SolucionPropuesta = Vacia;
25            ReiniciaAlgoritmo();
26        }
27    }
28    TiempoAlgoritmo = CalcularTiempoTranscurrido();
29 }
30
31 IncluirSolucionEnConjunto(ConjuntoSoluciones, SolucionMejora);
32 P->ConjuntoSoluciones = ConjuntoSoluciones;
33 Devolver(P);
```

# GRASP\_M

## ■ Optimizaciones:

### □ Linearización de la matriz de distancias:

$$d(i, j) = \text{vector}[i * n - (i * (i + 1) / 2) + j] \quad \forall 0 \leq i \leq j \leq n - 1$$

### □ Cálculo valor soluciones:

$$\text{Aportacion}(i) = \sum_{j=0}^{m-1} d(i, j) \quad \forall j \neq i, i \in \text{Solucion} \quad O(M)$$

$$\text{Solucion}(i) = \text{Solucion}(j) + \text{Aportacion}(i) - \text{Aportacion}(j)$$

$$\text{frente a ValorSolucion} = \sum_{i=0}^{m-1} \sum_{j=i+1}^{m-1} d(i, j) \quad O(M^2)$$

### □ Criba proceso de mejora:

$$\text{Aportacion}(i) = \sum_{j=0}^{m-1} d(i, j) < \min \left( \sum_{k=0}^{m-1} d(j, k) \right) \quad k \neq j, \forall j \in \text{Solucion}$$

# GRASP\_M

- Elección de parámetros:
  - Parámetros por defecto a priori:
    - MDP\_NSC = 5, MDP\_flags (MDP\_FSDO(a), MDP\_FSDOD(a), MDP\_FDOO(a), MDP\_FDOOD(a), MDP\_FEPS(a)), MDP\_MISM = 50, MDP\_PM = 0,05, MDP\_PREC = 6.
  - Parámetros tras experimentación:
    - El único cambio que mejora el comportamiento medio del algoritmo por defecto es la desactivación de la bandera MDP\_FEPS.

# GRASP\_M

## ■ Resultados experimentación: $GAP = \frac{Sol_{Alg} - Sol_{Best}}{Sol_{Best}}$

- Conjunto 8 problemas de tamaño 2000x200 .. 3000x600.

Tiempo límite (seg.)	Tiempo medio Sol (ms.)	GAP medio	% mejora (anterior fila)
100	58113,9	-0,00069930	0
300	186443,0	-0,00039213	44%
900	571023,1	-0,00012376	68% (82%)

- Resultados buenos, incluso mejores en algunos casos, con respecto a “Opticom Project” con tiempos de ejecución menores.
- Comportamiento correcto GRASP\_M en función del tiempo.

# EMP

- Esquema Metaheurístico Parametrizado:
  - Abstracción metaheurísticas.
  - Parámetros transicionales.
  - Aplicable a varios tipos de problemas con mínima adaptación.

```
1 Inicializar(S, ParamIni)
2 while (not ConficcionFin(S, ParamFin)) {
3     SS = Seleccionar(S, ParamSel)
4     SS1 = Combinar(SS, ParamCom)
5     SS2 = Mejorar(SS1, ParamMaj)
6     S = Incluir(SS2, ParamInc)
7 }
```



# EMP

## ■ Resultados experimentación 1:

- 7 Metaheurísticas con valores de referencia de parámetros EMP (GRASP, GA, SS, GRASP+GA, GRASP+SS, GA+SS, GRASP+GA+SS).
- Conjunto 6 problemas de tamaño 15x12 .. 150x45.
- 5 ejecuciones por problema y tiempo y GAP medio por problema.
- Mejores GAP medios con GRASP+SS y SS siendo la intensidad en la mejora y, por tanto los tiempos, menor en GRASP+SS.
- Incidencia importante del proceso de mejora en tiempos y resultados.

# EMP

## ■ Resultados experimentación 2:

- 4 Metaheurísticas híbridas con valores de parámetros EMP enfocados a la prueba del proceso de mejora.
- Mismo conjunto de problemas que el experimento 1.
- 5 ejecuciones por problema y tiempo y GAP medio por problema.
- Decrementos en la intensidad del proceso de mejora mejoran tiempo/iteraciones, incluso con eliminación de mejora en el inicio, y GAP medio.
- Comprobada incidencia en tiempos y resultados del buen calibrado del proceso de mejora.

# Conclusiones y trabajos futuros

## ■ Conclusiones:

- GRASP\_M parece ser un buen algoritmo para la resolución del problema MDP, al menos con los tamaños de problema tratados.
- EMP es una muy buena idea para el campo de las metaheurísticas y obtiene unos resultados correctos, al menos para problemas de tamaño pequeño.
- Posible desarrollo de hiperheurísticas para la selección automática de parámetros sin supervisión constante del usuario.
- A veces, sería conveniente crear algoritmos no necesariamente intentando mejorar o evolucionar algo ya existente, aunque luego termine por parecerse a lo ya existente.

# Conclusiones y trabajos futuros

## ■ Trabajos futuros:

- GRASP\_M podría ser mejorado con aplicación de ideas de otras metaheurísticas, mayor optimización y parametrización y un análisis e investigación más profundos sobre el problema MDP.
- Utilización del paralelismo para aumentar eficiencia de GRASP\_M.
- EMP podría ser mejorado en cuanto a una mayor optimización de sus procesos y una investigación más profunda del valor de sus parámetros para conseguir una mayor eficiencia.
- Desarrollo de hiperheurísticas para facilitar el refinamiento de los algoritmos.



**FIN**